REMARKS

Reconsideration and allowance are respectfully requested.

The previously-submitted figures are resubmitted with the appropriate labels. Approval is requested.

All claims 1-45 stand rejected under 35 U.S.C. 102 for anticipation based on newly-cited USP 6,477,638 to Gearty. This rejection is respectfully traversed.

To establish that a claim is anticipated, the Examiner must point out where each and every limitation in the claim is found in a single prior art reference. *Scripps Clinic & Research Found. v. Genentec, Inc.*, 927 F.2d 1565 (Fed. Cir. 1991). Every limitation contained in the claims must be present in the reference, and if even one limitation is missing from the reference, then it does not anticipate the claim. *Kloster Speedsteel AB v. Crucible, Inc.*, 793 F.2d 1565 (Fed. Cir. 1986). Gearty fails to satisfy this rigorous standard.

Gearty describes a tightly-coupled system, similar to that described in the background of the present application, where a coprocessor pipeline is synchronized with the main processor pipeline by passing signals with fixed timing from one pipeline to the other. Col. 6, lines 35-38 of Gearty describes that the CPU pipeline 160 and the FPU pipeline 162 are closely-coupled, (i.e., tightly-coupled), and therefore, an instruction image executes in the FPU pipeline 162 when a pure CPU instruction executes in the CPU pipeline 160. The same closely-coupled scheme is also used for FPU instructions, where an instruction image is maintained in the CPU pipeline 160 for FPU instructions executing in the FPU pipeline 162. See Col. 6, lines 55-65. Col. 8, lines 41 to 46 explain that the various latches provided between the CPU pipeline and the FPU pipeline are provided because of flight time issues between the CPU and the FPU. The background of

this application also identifies such timing issues as one of the problems exhibited by tightly-coupled synchronization schemes (see page 2, lines 13-26).

Gearty explains that the pipelines may slip out of synchronization. The pipelines are preferably only allowed to slip by one pipe stage, although in other embodiments, the slip may be more. Significantly, Gearty explains that the slip is "predetermined," which is a characteristic of tightly-coupled schemes that pass fixed timing signals between the pipelines. Due to the fixed timing, the amount of slip must be predetermined. This link between tightly-coupled schemes and a predetermined number of cycles of slip is further described at Col. 13, lines 48-55.

A problem with tightly-coupled schemes is that as the length of pipeline processors increases, it is more difficult to maintain pipeline synchronization because of signal propagation delays that make it difficult to ensure signals are passed between the pipelines with the required fixed timing. Independent claims 1 and 29 solve this problem using a token-based pipeline synchronization technique where at least one synchronizing queue couples a predetermined pipeline stage in one of the pipelines with a partner pipeline stage in the other of the pipelines. The predetermined pipeline stage and the partner pipeline stage transfer tokens to achieve a loosely-coupled synchronization scheme (see page 3, lines 23-29).

To help distinguish the claimed loosely-coupled scheme and Gearty's tightly-coupled scheme, independent claims 1 and 29 are amended to further clarify the claimed queues and the tokens. First, the synchronizing queue is defined as including a FIFO buffer having a predetermined plurality of entries. This amendment is supported for example by cancelled claim 26 and by the text of the application as originally filed, see for example page 11, lines 11 to 16. The presence of multiple entries in the queue results in variable timing in the transfer of tokens between the predetermined pipeline stage and the partner pipeline stage. If at the time a token is

placed in the queue, there are no further entries ahead of the token in the queue, then that token is transferred more quickly than if at the time the token is placed in the queue when there are multiple tokens ahead of that token in the queue. Second, the independent claims are amended to identify that the token includes a tag which uniquely identifies the coprocessor instruction to which the token relates. This language is supported for example by now cancelled claim 23, page 10, lines 13 to 26, and page 29, lines 10 to 17. Because there is no fixed timing for the transfer of information from the predetermined pipeline stage to the partner pipeline stage, the token uniquely identifies the coprocessor instruction to which the token relates so that the partner pipeline stage can react appropriately. Third, the amended claims emphasize that synchronization is achieved "without passing signals with fixed timing between the pipelines."

The loosely-coupled synchronization scheme recited in the independent claims provides a more flexible scheme that can be used in situations where the tightly-coupled synchronization scheme may be inoperable, such as in systems where the length of the pipelined processors is large, where signal delay propagation makes it difficult to use a tightly coupled synchronisation scheme, etc. Although the amount of slip between the pipelines is allowed to vary, the flexible loosely-coupled scheme ensures that the pipelines are correctly synchronized for crucial transfers of information.

Applicants respectfully disagree with the Examiner's assessment of claims 23, 26, 40 and 43. Regarding claims 23 and 40, it is not clear to which parts of Cols. 7 or 8 the Examiner is referring. Although Gearty refers to opcodes of instructions, they cannot be equated with the claimed "tag" that uniquely identifies the coprocessor instruction to which the token relates. In contrast, an opcode does not uniquely identify any particular instance of an instruction. For example, if the same instruction is executed twice in succession, those two instances of the

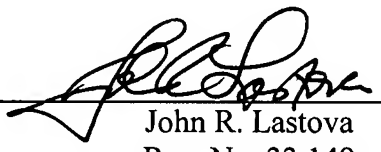
instruction have the same opcode. But in accordance with claims 1 and 29, the token placed within the synchronizing queue for each such instruction includes a different tag to enable those two instances of the same instruction to be uniquely identified.

Regarding the Examiner's comments with respect to claims 26 and 43, the section of the text being referred to by the Examiner relates to the construction of the CPU predecoder stage buffering mechanism described at Col. 4, lines 55-57. It has nothing to do with the synchronizing queues between pipelines. Gearty uses latches between the two pipelines. But a latch can only store a single discrete entry, and hence, cannot perform the operation of a FIFO buffer storing multiple discrete entries.

The application is in condition for allowance. An early notice to that effect is earnestly solicited.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: 

John R. Lastova
Reg. No. 33,149

JRL:sd
901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000
Facsimile: (703) 816-4100